

**EXAMINER'S AMENDMENT**

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.
2. Authorization for this examiner's amendment was given in a telephone interview with Jeffrey C. Hood, Reg.# 35198 on 03/18/2009.
3. **This listing of claims will replace all prior versions and listings of claims in the application:**
  - a. The following claims have been amended:
    1. (Currently Amended) A computer-accessible memory medium that stores program instructions for performing time-bounded execution of a program, wherein the program instructions are executable by a processor to perform:
      - 1) initiating a timed program execution process, wherein the timed program execution process is ~~operable~~ configured to execute a program in a time-bounded manner;
      - 2) initiating a timeout process, wherein the timeout process is ~~operable~~ configured to preempt the execution process to interrupt execution of the program;
      - 3) configuring a timeout event, wherein the timeout event is an event indicating a timeout condition for the program;

4) the timed program execution process performing a time-bounded execution of the program, comprising:

- a) determining and storing a rollback state for the program;
  - b) if the timeout event has not occurred, executing the program, wherein, during said executing, if the timeout event occurs,
  - c) the timeout process setting the timed program execution process to the rollback state, and disabling the timeout event; and
  - d) the timed program execution process resuming executing the program based on the rollback state with a timeout condition in preparation to perform a program exit procedure; and
  - e) performing the program exit procedure, thereby terminating execution of the program;
- 5) disabling the timeout event;
- 6) terminating the timeout process; and
- 7) terminating the timed program execution process.

2. (Previously Presented) The memory medium of claim 1, wherein e) further comprises:

clearing the rollback state.

3. (Previously Presented) The memory medium of claim 2, wherein the program instructions are further executable to perform:

iteratively performing said time-bounded execution of a plurality of programs.

4. (Previously Presented) The memory medium of claim 3, wherein said iteratively performing comprises:

for each of the plurality of programs, performing a) through e).

5. (Previously Presented) The memory medium of claim 4, wherein e) further comprises:

if the timeout event has occurred, storing an indication of a timeout condition; and

wherein the timed program execution process performing a time-bounded execution of the program comprises:

performing a) through e) if the timeout condition is not indicated.

6. (Previously Presented) The memory medium of claim 2, wherein the program instructions are further executable to iteratively perform:

setting a timeout event;

the timed program execution process performing a time-bounded execution of the program; and

disabling the timeout event.

7. (Previously Presented) The memory medium of claim 2, wherein the timeout process executes at a first priority level, and wherein the program instructions are further executable to perform:

setting an execution priority level of the timed program execution process to a second priority level, wherein the second priority level is below the first priority level.

8. (Previously Presented) The memory medium of claim 7, wherein said performing a time-bounded execution of the program further comprises:

storing an original execution priority level of the timed program execution process prior to said setting the execution priority level of the timed program execution process; and

wherein e) further comprises:

restoring the execution priority level of the timed program execution process to the original execution priority level.

9. (Previously Presented) The memory medium of claim 2,

wherein said initiating the timeout process comprises:

acquiring one or more resources for the time-bounded execution of the program; and

initializing one or more resource managers for the one or more resources;  
and

wherein said terminating the timeout process comprises:

releasing the one or more resources for the time-bounded execution of the program; and

un-initializing the one or more resource managers for the one or more resources.

10. (Previously Presented) The memory medium of claim 9,

wherein said initiating the timeout process further comprises:

creating each of the one or more resource managers; and  
wherein said terminating the timeout process further comprises:

deleting each of the one or more resource managers.

11. (Previously Presented) The memory medium of claim 9,

wherein said initiating the timeout process further comprises:

enabling each of the one or more resource managers; and

wherein said terminating the timeout process further comprises:

disabling each of the one or more resource managers.

12. (Previously Presented) The memory medium of claim 9, wherein said one or more resources comprises:

one or more memory pools for memory allocations during the time-bounded execution of the program.

13. (Previously Presented) The memory medium of claim 9,

wherein, prior to a), the program instructions are further executable to perform:

storing a respective resource state for each of the one or more resources;

and

enabling each of the one or more resource managers.

wherein e) further comprises:

if the timeout event has occurred,

restoring the respective resource state for each of the one or more resources; and

disabling each of the one or more resource managers.

14. (Previously Presented) The memory medium of claim 1, wherein said initiating the timeout process is performed during said executing.

15. (Previously Presented) The memory medium of claim 1,  
wherein the program comprises one or more sub-programs; and  
wherein b) further comprises:

performing a) through c) for each of the one or more sub-programs.

16. (Previously Presented) The memory medium of claim 1, wherein said initiating the timeout process is performed by the timed program execution process.

17. (Previously Presented) The memory medium of claim 1, wherein the rollback state comprises:

a rollback point, comprising an execution point in the program, and  
an execution state of the timed program execution process at the rollback point.

18. (Previously Presented) The memory medium of claim 1, wherein the program comprises a subprogram, and wherein said executing the program further comprises:

8) initiating another timeout process;

9) configuring another timeout event;

10) the timed program execution process performing a time-bounded execution of the subprogram, comprising:

f) determining and storing another rollback state for the subprogram;

g) if the another timeout event has not occurred, executing the subprogram, wherein, during said executing the subprogram, if the another timeout event occurs,

h) the another timeout process setting the timed program execution process to the another rollback state, and disabling the another timeout event; and

i) the timed program execution process resuming executing the subprogram based on the another rollback state with another timeout condition; and

j) performing a subprogram exit procedure, thereby terminating execution of the subprogram;

11) disabling the another timeout event; and

12) terminating the another timeout process.

19. (Previously Presented) The memory medium of claim 18, wherein j) further comprises e).

20. (Previously Presented) The memory medium of claim 1, wherein the program instructions are further executable to perform time-bounded execution of another program substantially in parallel with the time-bounded execution of the program by:

8) initiating another timeout process;

9) configuring another timeout event;

10) the timed program execution process performing a time-bounded execution of the another program, comprising:

f) determining and storing another rollback state for the another program;

g) if the another timeout event has not occurred, executing the another program, wherein, during said executing the another program, if the another timeout event occurs,

h) the another timeout process setting the timed program execution process to the another rollback state, and disabling the another timeout event; and

i) the timed program execution process resuming executing the another program based on the another rollback state with another timeout condition; and

j) performing another program exit procedure, thereby terminating execution of the other program;

11) disabling the another timeout event; and

12) terminating the another timeout process.

21. (Previously Presented) The memory medium of claim 1, wherein said executing the program further comprises:

receiving a disable request from the program to disable the rollback state;

disabling the rollback state in response to said disable request;

receiving an enable request from the program to enable the rollback state;

enabling the rollback state in response to said enable request; and

updating the rollback state for the program.

22. (Previously Presented) The memory medium of claim 1,  
wherein the program comprises a machine vision application.

23. (Previously Presented) The memory medium of claim 1,  
wherein the program comprises a numerical analysis application.

24. (Previously Presented) The memory medium of claim 1,



wherein the program comprises a text-based program.

25. (Previously Presented) The memory medium of claim 1, wherein the program comprises a graphical program.

26.-27. (Canceled)

28. (Currently Amended) A system for performing time-bounded execution of a program, wherein the system comprises:

a processor; and

a memory coupled to the processor, wherein the memory stores program instructions executable by the processor to:

1) initiate a timed program execution process, wherein the timed program execution process is ~~operable~~ configured to execute a program in a time-bounded manner;

2) initiate a timeout process, wherein the timeout process is ~~operable~~ configured to preempt the execution process to interrupt execution of the program;

3) set a timeout event, wherein the timeout event is an event indicating a timeout condition for the program;

wherein the timed program execution process is ~~operable~~ configured to:

4) perform a time-bounded execution of the program, wherein, in performing the time-bounded execution, the timed program execution process is ~~operable~~ configured to:

a) determine and store a rollback state for the program;

b) if the timeout event has not occurred, execute the program, wherein, during said executing, if the timeout event occurs,

c) the timeout process is ~~operable~~ configured to set the timed program execution process to the rollback state, and disable the timeout event; and

d) the timed program execution process is ~~operable~~ configured to resume executing the program based on the rollback state with a timeout condition in preparation to perform a program exit procedure; and

e) perform the program exit procedure, thereby terminating execution of the program;

wherein the program instructions are further executable by the processor to:

5) disable the timeout event;

6) terminate the timeout process; and

7) terminate the timed program execution process.

29. (Currently Amended) A computer system for performing time-bounded execution of a program, comprising:

1) means for initiating a timed program execution process, wherein the timed program execution process is ~~operable~~ configured to execute a program in a time-bounded manner;

2) means for initiating a timeout process, wherein the timeout process is ~~operable~~ configured to preempt the execution process to interrupt execution of the program;

3) means for setting a timeout event, wherein the timeout event is an event indicating a timeout condition for the program;

4) means for the timed program execution process performing a time-bounded execution of the program, comprising:

a) means for determining and storing a rollback state for the program;

b) means for: executing the program if the timeout event has not occurred, wherein, during said executing, if the timeout event occurs,

c) the timeout process setting the timed program execution process to the rollback state, and disabling the timeout event; and

d) the timed program execution process resuming executing the program based on the rollback state with a timeout condition in preparation to perform a program exit procedure; and

e) means for performing the program exit procedure to terminate execution of the program;

5) means for disabling the timeout event;

6) means for terminating the timeout process; and

7) means for terminating the timed program execution process.

30. (Canceled).

#### ***Reason for Allowance***

4. The following is an examiner's statement of reasons for allowance:

"LabVIEW Real-Time module User Manual" teaches executing program threads in a time-bounded manner with time-out condition. A user can execute a thread as desired with time-out condition to monitor the program execution in real-time within a time constrain.

Fuchs et al. (US Patent No. 5530802) teaches a monitoring program execution and for error condition. System can store checkpoint for stable state and rollback to a stable state and disable the error condition to bypass the system failure.

The cited prior art references do not teach c) the timeout process setting the timed program execution process to the rollback state, and disabling the timeout event; and d) the

timed program execution process resuming executing the program based on the rollback state with a timeout condition in preparation to perform a program exit procedure; and e) performing the program exit procedure, thereby terminating execution of the program; 5) disabling the timeout event; 6) terminating the timeout process; and 7) terminating the timed program execution process.

Although the specification does not provide antecedent bases for the limitation "computer-accessible memory medium" (as recited in independent claim 1 and dependent claims 2-25) applicant's specification page 12, lines 27-28 through page 13, lines 1-11 discloses:

Memory Medium--Any of various types of memory devices or storage devices. The term "memory medium" is intended to include an installation medium, e.g., a CD-ROM, floppy disks 104, or tape device; a computer system memory or random access memory such as DRAM, DDR RAM, SRAM, EDO RAM, Rambus RAM, etc.; or a non-volatile memory such as a magnetic media, e.g., a hard drive, or optical storage. The memory medium may comprise other types of memory as well, or combinations thereof. In addition, the memory medium may be located in a first computer in which the programs are executed, or may be located in a second different computer which connects to the first computer over a network, such as the Internet. In the latter instance, the second computer may provide program instructions to the first computer for execution. The term "memory medium" may include two or more memory mediums which may reside in different locations, e.g., in different computers that are connected over a network.

It would reasonably be interpreted by one of the ordinary skill in the art that the teaching as identified above constitutes "computer-accessible memory medium" and limits the medium to a computer-readable storage medium.

Claim 29 is directed to a Computer system for performing the steps of the execution invoking 112, 6th paragraph. The computer system environment clearly shows being executed in combination of software and hardware as shown in Figure 2A and 2B. Therefore claim 29 is statutory subject matter and do not have any 101 issue.

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

### ***Conclusion***

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ABDULLAH AL KAWSAR whose telephone number is (571)270-3169. The examiner can normally be reached on 7:30am to 5:00pm, EST.

7. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng Ai T. An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

8. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/VAN H NGUYEN/  
Primary Examiner, Art Unit 2194

/Abdullah-Al Kawsar/  
Examiner, Art Unit 2195